Contribution ID: **18** Type: **not specified**

Resource Efficient and Low Latency GNN-based Particle Tracking on FPGA

Monday, 3 October 2022 16:15 (15 minutes)

Charged particle tracking is important in high-energy particle physics. For CERN Large Hadron Collider (LHC), tracking algorithms are used to identify the trajectories of charged particles created in the collisions. The existing tracking algorithms are typically based on the combinatorial Kalman filter where the complexity increases quadratically with the number of hits. The poor scalability issue will be exacerbated when the beam intensities are expected to increase dramatically. Therefore, new tracking algorithms based on Graph Neural Networks (GNNs) are introduced to enhance the scalability of particle tracking tasks. These GNN algorithms are implemented on Field Programmable Gate Arrays (FPGAs) to meet the strict latency requirement of fast particle tracking. However, the previous design on Xilinx Virtex UltraScale+ VU9P FPGA can only accommodate a small GNN (28 nodes / 56 edges) due to the significant resource requirement of complex graph processing patterns. A collision event (660 nodes / 1320 edges) needs to be partitioned into smaller sub-graphs to fit the GNN processing to VU9P FPGA. Dividing a collision event into smaller sub-graphs could cause a higher possibility of missing important trajectories between sub-graphs.

In this work, we introduce a resource efficient and low latency architecture to accelerate large GNN processing on FPGA. This design leverages the GNN processing patterns and trajectory data properties to significantly improve the parallelism and computation throughput. We propose a highly parallel architecture with configurable parameters for users to adjust latency, resource utilization, and parallelism. A customized data allocation is used to address the irregular processing patterns and attain high processing parallelism. We further exploit the properties of trajectories between inner and outer detector layers, and reduce the unnecessary dependencies and edges in the graph.

The design is synthesized using hls4ml and implemented on Xilinx Virtex UltraScale+ VU9P FPGA. The proposed design can support a graph of size 660 nodes and 1560 edges with Initialization Interval of 200 ns.

Primary authors: LAI, Bo-Cheng; HUANG, Shi-Yu

Co-authors: ELABD, Abdelrahman; DUARTE, Javier (Univ. of California San Diego (US)); HU, Jin-Xuan; NEUBAUER, Mark (Univ. Illinois at Urbana Champaign (US)); ATKINSON, Markus (Univ. Illinois at Urbana Champaign (US)); HAUCK, Scott; HSU, Shih-Chieh (University of Washington Seattle (US)); RAZAVIMALEKI, Vesal (Univ. Illinois at Urbana-Champaign (US))

Presenters: LAI, Bo-Cheng; HUANG, Shi-Yu

Session Classification: Contributed Talks